SSRG International Journal of VLSI and Signal Processing
ISSN: 2394–2584 / https://doi.org/10.14445/23942584/IJVSP-V10I2P102

Volume 10 Issue 2, 12-16, May-Aug 2023

Original Article

Design and Implementation of Flight Control System on FPGA

P. Anjali[1], Anushka L Naik[2], N. S. Chinmai[3], P. J. Febha[4], S. Jamuna[5]

[1,2,3,4,5]*Electronics & Communication Engineering, Dayananda Sagar College of Engineering, Karnataka, India.*

Received: 27 April 2023 Revised: 02 June 2023 Accepted: 16 June 2023 Published: 30 June 2023

Abstract - *A flight control system (FCS) is a major component of an aircraft. It allows the pilot to monitor the forces of flight, direction and attitude of the aircraft during flight. A flight control system regulates the necessary inputs to manipulate the control surfaces for the pilot to monitor the aircraft. They include cockpit controls, connecting linkages, aircraft engine controls, and necessary operating mechanisms. In a conventional mechanical flight control system, a system breakdown can be catastrophic as there are no other alternative backups. It is, therefore, necessary to formulate a system that is developed with a considerable amount of redundancy & fault tolerance. Thus, the main objective of this project is to design and implement a flight control system using a field programmable gate array (FPGA). The performance of the FPGA-based FCS system is better than that of the conventional microcontroller and the DSP chip-based UAV flight control system, and it has several useful applications and benefits. The entire flight control system is divided into four modules navigation control module, flight control module, sensor driver module, and Avalon bus control module.*

Keywords - *Unmanned Aerial Vehicle (UAV), Flight Control System (FCS), Field Programmable Gate Array (FPGA), GPS (Global Positioning System).*

1. Introduction

The FCS is an essential part of an aircraft that gives pilots control and maneuverability while in the air. The FCS uses various sensors and actuators to interpret pilot inputs and adjust the aircraft's control surfaces, such as ailerons, elevators, and rudder, to achieve the desired flight parameters [1]. The FCS system is essential for maintaining an airplane's stability and control, as it detects inputs from sensors and automatically adjusts control surfaces to stabilize the aircraft and ensure a predictable response to pilot commands. Pilots can manipulate the control inputs to adjust the aircraft's pitch, roll, and yaw, enabling them to manage the aircraft's attitude [3]. Many modern planes have autopilot integrated into their FCS, which can use inputs from GPS and other sensors to navigate and maintain the desired settings, allowing pilots to take a break from continuous manual control. FBW, short for Fly-By-Wire System, is an electronic substitute for traditional mechanical systems connecting cockpit controls with aircraft control surfaces. In FBW systems, electronic signals are used to transmit the pilot's instructions to the onboard computers, which decide how to move the control surfaces. FBW offers several benefits, such as increased stability and control and reduced weight and maintenance requirements [4]. In order to ensure that the aircraft can continue to operate even if a component fails, safety procedures that identify mistakes or system abnormalities are built into flight control systems. This involves using multiple sensors and actuators, so the system can switch to a backup if one component malfunctions. The FCS could have functionalities to ensure the airplane does not surpass its safe operational boundaries. Flight envelope protection features keep an eye on a number of variables, including the plane's speed and angle of attack, to prevent dangerous manoeuvres or stall situations. An aircraft's safe and efficient operation depends on a flight control system. By incorporating advanced technology alongside automation systems as well as taking into account inputs from the pilot, can maintain stability and control across all stages of flight [6]. The process of designing and implementing a flight control system utilizing an FPGA necessitates the integration of several elements and algorithms to achieve stable and precise aircraft control. The system specifications must be defined, which includes the desired flight dynamics, control modes, sensor inputs, and desired outputs. Interfacing the sensors with the FPGA, implementing control algorithms for stabilization and navigation, and interfacing the actuators with the FPGA are all essential steps in the process. The interconnections, data busses, and control interfaces within the FPGA must be designed and implemented to form a complete flight control system [8]. A testing and verification plan to validate the functionality and performance of the FCS must be developed, and the system's performance must be analysed and optimized as needed. To ensure safety and reliability, redundancy measures such as redundant sensors, voting logic, and fault detection mechanisms can be implemented.

It is important to note that designing a flight control system requires a profound understanding of control theory, flight dynamics, and FPGA programming. In order to guarantee the security and dependability of the FCS, it is advised to seek the advice of subject matter specialists and adhere to accepted industry norms and rules [10]. The flight control system comprises four modules: navigation control module, flight control module, sensor driver module, and Avalon bus control module. The sensor driver module's primary responsibility is to control the system's sensors, which are used for data collection and pre-processing. The pre-processed data is subsequently entered into the navigation control IP core and flight control IP core through the bus and any applicable communication protocol rules. Second, there are two primary functions for the navigation control IP core module. The initial objective is to thoroughly process the data gathered by the accelerometer, magnetoresistance metre, gyroscope, and other sensors to obtain stable flight altitude information. In order to gain exact navigational data regarding the flight, the second step entails processing data gathered by SINS, GPS, airspeed metres, and altimeter using the extended Kalman filters technique. Thirdly, the flight control IP core module will mainly compute the navigation data and ground control signals. The results and control signals are then output to the flight servo mechanism using the matching algorithm. The drive signal regulates the brushless DC motor, which in turn regulates the flight path and altitude. The final component that sets the communication protocol for the whole flight control system is the Avalon bus control module. This guarantees reliable and fast data transfer between each module, ultimately enabling exact aircraft control [12]. The Field Programmable Gate Array (FPGA) technology enables parallel processing, allowing for multiple bits to be processed simultaneously. It also allows for a large number of gates to be integrated into a single chip, which is essential for reducing the volume of the system. Additionally, using FPGA technology can result in cost savings. Code parallelization can increase processing speed in both FPGA and other hardware components, and the software employed in the system architecture can substantially impact the total performance. FPGAs are frequently employed as the principal architecture platform in flight control systems in contemporary flight development because they provide quick and effective answers while validating the concept and saving time. The FPGA/DSP-based controller is the ideal approach utilised for FCS implementation because of its low power consumption, quick reaction, small size, and compatibility with applications. Other approaches include PID Controller, a combination of PID Controller with Kalman filter, and fuzzy logic Controller [15]. The flight control module will be coded using Verilog language and implemented on an FPGA.

2. Related Works

Blake Fuller, Jonathan Kok et al. worked on [1] developing a small UAV capable of autonomous operation despite constraints such as limited processing capabilities, power, and size. The team established connectivity between the onboard flight computer and the FPGA-based path planner using the MAVLink protocol.

The four techniques for autonomous flight control systems, including fuzzy logic, PID controller, FPGA/DSP technology, and Kalman filter, were reviewed in [3]. The resulting system exhibited characteristics suitable for unmanned aviation applications.

Using FPGA embedded systems, Noe' Monterrosa et al.'s research paper [4] from September 2017 described a flight controller based on a state machine technique. The controller responded quickly to stimuli from sensors and was adaptable in calibration.

A dual-core UAV flight control system based on an FPGA chip's high-speed parallel processing capability and Altera's SOPC development tool Qsys [6] was presented by Kun Zhang et al. in April 2019. The system was highly integrated, robust, and had strong refactoring capabilities.

The research [7] offers a power line identification system based on field-programmable gate arrays (FPGAs) and stereo vision technologies. The system's goal is to detect power lines in real-time, correctly and effectively.

In the research [8], they provide a field-programmable gate array (FPGA) based video image monitoring system for bus lanes. The system's goals include providing real-time video analysis, processing, and efficient bus lane monitoring.

The paper [10] provides a control system that makes use of an Extended Kalman Filter (EKF) as an observer for the health monitoring of a metal-polymer hybrid soft actuator. The control structure's goal is to monitor the actuator's state of health and find any potential flaws or irregularities.

3. Proposed Work

3.1. Block Diagram

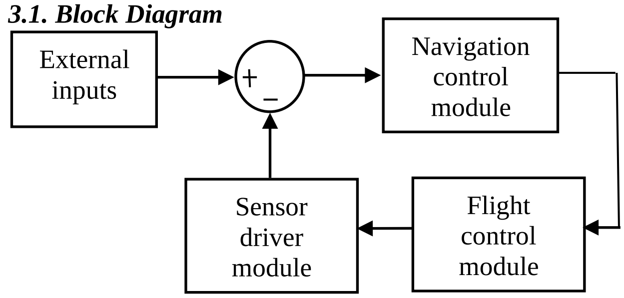

Fig. 1 Block diagram

3.2. Methodology

The entire design was designed using Matlab Simulink. The motor or propeller block with voltage as the input was designed first. It produced thrust and torque as the output. The outputs of the motor or propeller block were fed as input to the rotational dynamic block. Since integration is an easy operation in Simulink, the output from the rotational dynamic block, which is angular acceleration, was given as input to an integrator which gave rotational velocity. The rotational velocity on further integrating gave theta, which is the angle.

The force which was taken as the output from the motor or propeller block was fed as an input to the linear dynamic block, which gave linear acceleration as the output. Linear acceleration on further integrating gave linear velocity, which was made to pass an integrator block and give us the linear position. A drag disturbance block was created with wind gusts as the input along the three axes to understand better the disturbances that will affect the flight controller. Disturbance, which was the disturbance block's output, was used as the linear dynamic block's input to produce an exact linear position. Since the navigation control module can only accept voltage as an input, we used a human control block to convert attitude variables such as yaw, pitch, and throttle to voltage. The values for yaw, pitch, roll, and throttle are fed manually. The linear position is fed back to the summer, which performs the desired calculation required for flight movements. The PID controller block quickly implements all the corrections required for the block.

The code we used in Matlab Simulink is converted into Verilog and an extra 8:1 mux to see the outputs separately in FPGA.

3.3. Hardware and Software Requirements

3.3.1. Hardware Requirements

FPGAs play a significant role in system development. The four fundamental modules in this project are designed using the Matlab Simulink software. The flight control module is also programmed in Verilog and implemented on an FPGA.

3.3.2. Software Requirements

Throughout the design process of this project, the software program Simulink is being used, specifically the most recent version, R2022a. Matlab permits users to write code and construct blocks. The codes are written on the command window, while the blocks are available in Simulink's libraries. Simulink, Simulink 3D analyzer, and DSP system tools are a few of the libraries. The ability to provide desired inputs to the code is one of Simulink's main benefits.

The flight control module is designed in Verilog language and simulated using Model Sim.

For better waveform simulation, visual editing of logic circuits, and hardware description, Quartus Prime has an implementation of VHDL and Verilog. This software is used to get RTL view and implement it on hardware.

4. Results and Discussion

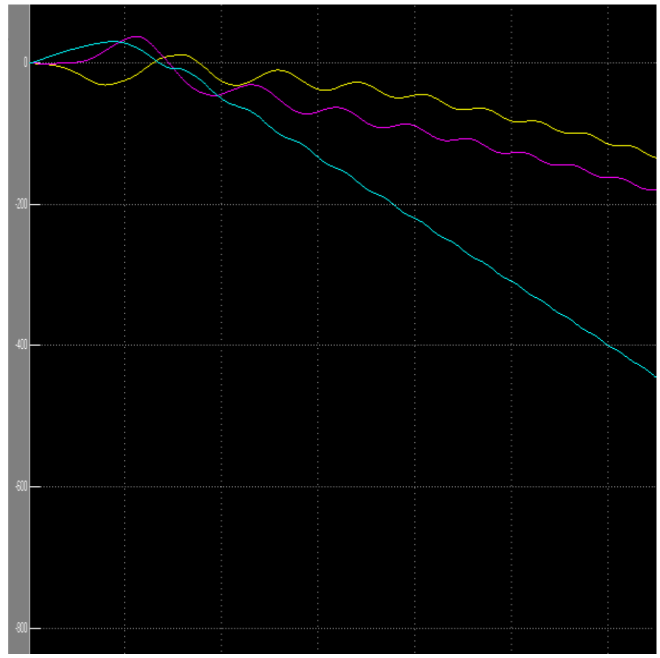

Fig. 2 Output waveform (positive drag)

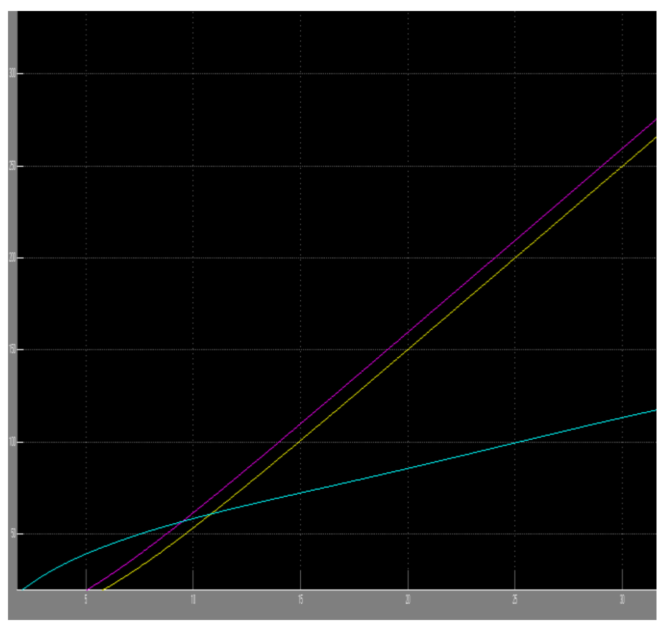

Fig. 3 Output waveform (negative drag)

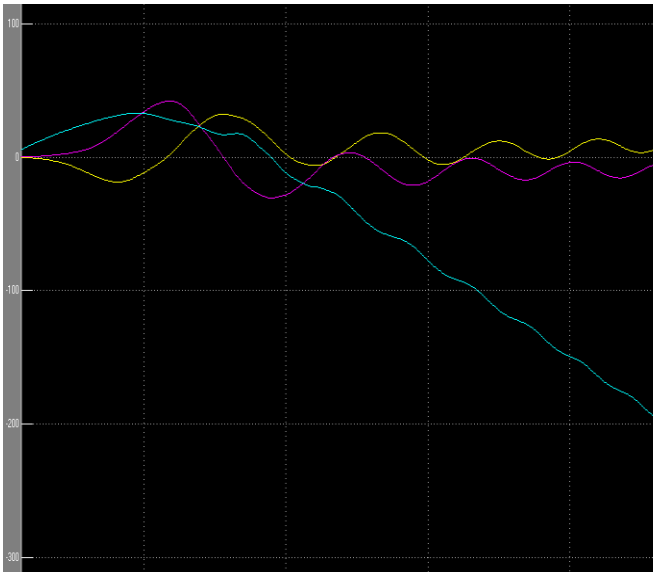

Fig. 4 Output waveform (zero drag)

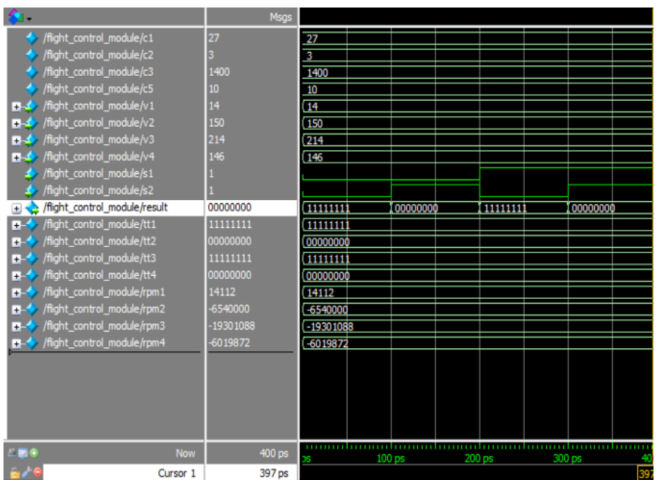

Fig. 5 Model Sim output waveform (case I)

The simulated output of the FCS is shown in Figures 2,3, and 4, where the flying position is visible under three different conditions. The first type of drag is a positive drag, where the linear position is in a downward or negative direction.

The second is the negative drag, which occurs when the FCS is positioned positively or upward. The third one is zero drag. In that, the linear position is more towards the zero axis.

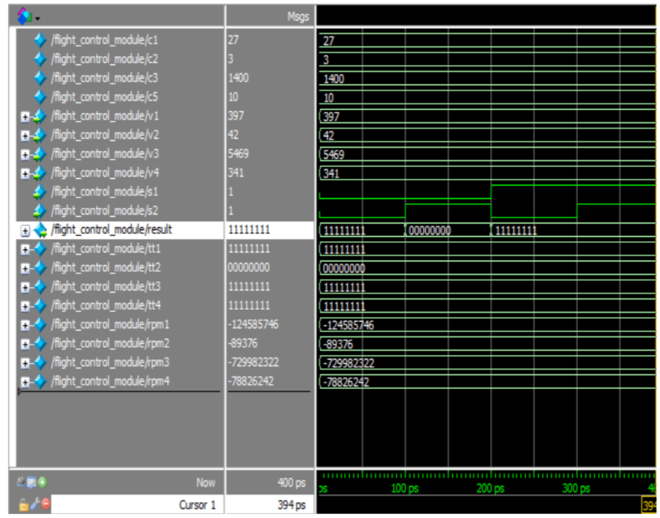

Fig. 6 Model sim output waveform (case II)

The flight control module's simulated output with inputs v1=14V, v2=150V, v3=214V, and v4=146V is depicted in Figure 4 by ModelSim. Here, we varied the voltages and examined the output in the throttles. Figure 6 shows the flight control module's simulated output with inputs v1= 397V, v2=42V, v3= 5469V, v4= 341.

5. Conclusion

Unlike the FCS, which was used in earlier days like the mechanical and microcontroller based, the designed one is more efficient. The mechanical-based flight control systems were found to be heavy. They also required careful routing of cables. Since the routing had to be done cautiously, there was a need for redundant backup, which increased the weight of the FCS. Microcontroller-based flight control systems have less capacity for onboard processing of autonomous real-time image processing for path planning and avoidance.

References

[1] Blake Fuller et al., "Hardware Design and Implementation of a MAVLink Interface for an FPGA – Based Autonomous UAV Flight Control System," *Australasian Conference on Robotics and Automation Association*, pp. 1-6, 2014. [Google Scholar] [Publisher Link]

[2] Shilpa Kedari et al., "Real Time Wireless Communication between Quadcopter and Android in Agriculture Field – A Review," *SSRG International Journal of Computer Science and Engineering*, vol. 2, no. 12, pp. 15-18, 2015. [CrossRef] [Publisher Link]

[3] Banwari Lal Sharma, Narendra Khatri, and Abhishek Sharma, "An Analytical Review on FPGA Based Autonomous Flight Control System for Small UAVs," *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 1369-1372, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[4] Noe' Monterrosa et al., "Design, Development, and Implementation of a UAV flight controller based on a state machine approach using an FPGA embedded system," *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1-8, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] K.Hemachandran et al., “Method of Improving Power Quality Using FPGA with SAPF in Power Systems,” *SSRG International Journal of Electrical and Electronics Engineering*, vol. 2, no. 10, pp. 8-12, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Kun Zhang et al., “Design and Implementation of a Dual – IP Core UAV flight Control System based on Qsys,” *Procedia Computer Science*, vol. 166, pp. 180-186, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Xiao Zhou, Xiaoliang Zheng, and Kejun Ou, “Power Line Detect System Based on Stereo Vision, and FPGA,” *IEEE 2nd International Conference on Image, Vision, and Computing*, pp. 715-719, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Li Jinghong, Tian Yanan, and Xu Xiujian, “Design and Implementation of Bus Lane Video Image Monitor System Based on FPGA,” *29th Chinese Control and Decision Conference*, pp. 4868-4872, 2017 [CrossRef] [Google Scholar] [Publisher Link]

[9] M. R. Ezilarasan, and J. Brittopari, "An Efficient FPGA-Based Adaptive Filter for ICA Implementation in Adaptive Noise Cancellation," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 10, no. 1, pp. 117-127, 2023. [CrossRef] [Publisher Link]

[10] Manuel Schimmack, Benedikt Haus, and Paolo Mercorelli, “An Extended Kalman Filter as an Observer in a Control Structure for Health Monitoring of a Metal-Polymer Hybrid Soft Actuator,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1477-1487, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Axel Barrau, and Silvère Bonnabel, “The Invariant Extended Kalman Filter as a Stable Observer,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797-1812, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Meng Zhang et al., “PID Passivity-Based Control of Port-Hamiltonian Systems,” *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1032-1044, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[13] Arunjyothi Eddla, and Venkata Yasoda Jayasree Pappu, “FPGA based Matched Filter Design using Modified Masking Signal Generator,” *International Journal of Engineering Trends and Technology*, vol. 70, no. 10, pp. 1-7, 2022. [CrossRef] [Publisher Link]

[14] V.M. Konyushko, “Features of UAV Control Systems Assessing,” *IEEE 2nd International Conference Actual Problems of Unmanned Air Vehicles Developments Proceedings*, pp. 199-202, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[15] Zhengrong Li et al., “Toward Automated Power Line Corridor Monitoring Using Advanced Aircraft Control and Multisource Feature Fusion,” *Journal of Field Robotics*, vol. 29, no. 1, pp. 4-24, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[16] Jon Kok, Luis Felipe Gonzalez, and Neil Kelson, “FPGA Implementation of an Evolutionary Algorithm for Autonomous Unmanned Aerial Vehicle On-Board Path Planning,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 2, pp. 272-281, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[17] Pradeep R. Fernando et al., “Customizable FPGA IP Core Implementation of a General-Purpose Genetic Algorithm Engine,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 133-149, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[18] François C. J. Allaire et al., “FPGA Implementation of Genetic Algorithm for UAV Real-Time Path Planning,” *Unmanned Aircraft Systems*, vol. 54, no. 1-3, pp. 495–510, 2009. [CrossRef] [Google Scholar] [Publisher Link]